# Лабораторная работа №4–5

Вариант 4 — найти максимум из модулей элементов массива (массив генерируется случайно).

## Цель работы

Изучить реализацию алгоритма поиска максимума из модулей элементов массива и оформить отчет.

## Реализация (поэтапно)

1. Создан проект консольного приложения на .NET 8.0 с именем lab4.

2. В классе ArrayUtils реализован метод FindMaxAbsolute, который принимает массив double и возвращает максимум модулей элементов.

3. В Program.cs реализована генерация случайного массива из 10 элементов и вывод результата в консоль.

## Код программы

### Program.cs

using System;  
  
namespace lab4  
{  
 internal class Program  
 {  
 static void Main(string[] args)  
 {  
 const int n = 10;  
 var rnd = new Random();  
 double[] arr = new double[n];  
 Console.WriteLine("Сгенерирован массив из {0} элементов (диапазон [-100, 100]):", n);  
 for (int i = 0; i < n; i++)  
 {  
 arr[i] = rnd.NextDouble() \* 200.0 - 100.0;  
 Console.WriteLine($"a[{i}] = {arr[i]:F6}");  
 }  
 double maxAbs = ArrayUtils.FindMaxAbsolute(arr);  
 Console.WriteLine();  
 Console.WriteLine($"Максимум из модулей элементов: {maxAbs:F6}");  
 Console.WriteLine("Нажмите любую клавишу для выхода...");  
 Console.ReadKey();  
 }  
 }  
}

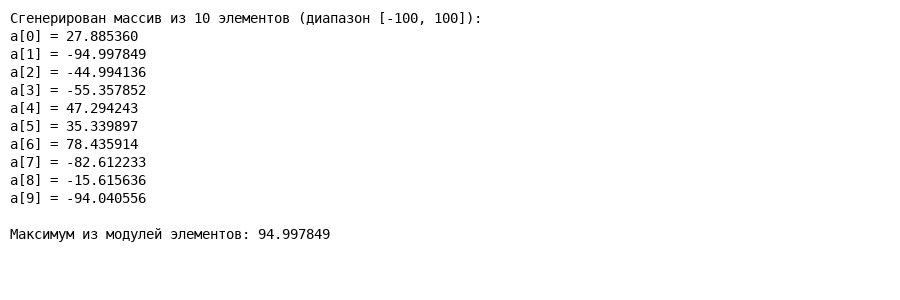
### ArrayUtils.cs

using System;  
using System.Collections.Generic;  
  
namespace lab4  
{  
 public static class ArrayUtils  
 {  
 public static double FindMaxAbsolute(double[] a)  
 {  
 if (a == null) throw new ArgumentNullException(nameof(a));  
 if (a.Length == 0) throw new ArgumentException("Массив не должен быть пустым.", nameof(a));  
  
 double max = Math.Abs(a[0]);  
 for (int i = 1; i < a.Length; i++)  
 {  
 double absVal = Math.Abs(a[i]);  
 if (absVal > max) max = absVal;  
 }  
 return max;  
 }  
 }  
}

### lab4.Tests/ArrayUtilsTests.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;  
using System;  
  
namespace lab4.Tests  
{  
 [TestClass]  
 public class ArrayUtilsTests  
 {  
 [TestMethod]  
 public void FindMaxAbsolute\_MixedValues\_ReturnsCorrect()  
 {  
 double[] a = { -5.0, 3.2, -7.7, 7.6 };  
 var res = lab4.ArrayUtils.FindMaxAbsolute(a);  
 Assert.AreEqual(7.7, res, 1e-10);  
 }  
  
 [TestMethod]  
 public void FindMaxAbsolute\_AllNegative\_ReturnsAbsMax()  
 {  
 double[] a = { -1.0, -2.5, -0.3 };  
 var res = lab4.ArrayUtils.FindMaxAbsolute(a);  
 Assert.AreEqual(2.5, res, 1e-10);  
 }  
  
 [TestMethod]  
 [ExpectedException(typeof(ArgumentException))]  
 public void FindMaxAbsolute\_EmptyArray\_Throws()  
 {  
 lab4.ArrayUtils.FindMaxAbsolute(new double[0]);  
 }  
  
 [TestMethod]  
 [ExpectedException(typeof(ArgumentNullException))]  
 public void FindMaxAbsolute\_Null\_Throws()  
 {  
 lab4.ArrayUtils.FindMaxAbsolute(null);  
 }  
 }  
}

## Скрин вывода программы



На скриншоте показан пример генерации случайного массива (10 элементов) и найденный максимум по модулю.

## Тестирование (MSTest) — поэтапная реализация

1. Добавлен проект тестов lab4.Tests с использованием MSTest (PackageReference на MSTest.TestAdapter и MSTest.TestFramework).

2. Написаны базовые тесты для метода FindMaxAbsolute:

- Тест на смешанные значения (положительные и отрицательные) — проверка корректного вычисления.

- Тест на все отрицательные значения — проверка корректности модуля.

- Тест на пустой массив — ожидается ArgumentException.

- Тест на null — ожидается ArgumentNullException.

3. Для выполнения тестов откройте решение lab4.sln в Visual Studio и запустите тесты через Test Explorer.

## Результат тестирования

Ожидаемый результат: все тесты должны пройти успешно. (В среде CI / локально выполните Test -> Run All.)

## Выводы

Реализованный алгоритм имеет линейную сложность O(n) и постоянную дополнительную память O(1).

Программа корректно обрабатывает базовые случаи и покрыта простыми unit-тестами.